

INFORMAL COMMUNICATION: Please do not put in the file

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE (USPTO)

Serial Number	10/693,854
Confirmation Number	3939
Filing Date	Oct 24, 2003
Title of Application	Programming Interface for a Computer Platform
First Named Inventor	Robert A. Relyea
Assignee	Microsoft Corporation
Group Art Unit	2192
Examiner	Ben C Wang
Attorney Docket Number	MS1-1780US
Nature of this Document	Informal Communication in Preparation for Scheduling an Examiner Interview

To: Examiner Wang
Fax: (571) 270-2240
Phone: (571) 270-1240

From: Rob Peck
Lee & Hayes, PLLC
421 W. Riverside Avenue, Suite 500
Spokane, WA 99201
robp@leehayes.com
(Tel. 509-324-9256; Fax 509-323-8979)

Dear Examiner Wang:

[0001] This communication provides an agenda for an interview of this matter. My assistant will be contacting you to schedule an interview. If possible, I would like to schedule the interview between September 29th and October 7th. If you would prefer to schedule the interview, then please contact my assistant or me directly. Our contact info is on the signature

Rev. Date: 04/01/08
Rev. By: KDB

INFORMAL COMMUNICATION: Please do not put in the file

page of this document. Thank you in advance for talking with me about this matter.

Interview Agenda:

- Discussion of current § 101 rejections;
- Discussion of exemplary differences between the application/claims and the cited references; and
- Discussion of proposed amendments

Section 101

[0002] I would like to confirm that you will withdraw the current § 101 rejections in light of the amendments proposed herein. Specifically, I propose amending claims 1 and 17 to recite *storage* media.

[0003] If you will not withdraw the current §101 rejections if the proposed amendments are submitted, I would like to discuss suggestions you may have for additional amendments.

Exemplary Differences

[0004] The Application describes a programming interface having a plurality of groups of services that enable a programmer to generate a user interface,

[0005] For example, claim 1 recites (in part):

INFORMAL COMMUNICATION: Please do not put in the file

generating graphical objects using a first group of services;

formatting content using a second group of services,
wherein the second group of services arrange the graphical objects,

creating components of the graphical objects using a third group of services

[0006] In contrast, the primary reference (Smith) describes a programming interface which enables a programmer to specify applications such as web services. The programming interface of Smith also includes a plurality of groups of services. Services that you cited include those enabling drawing of objects and security. Also, Smith mentions that the use of XML and HTML.

[0007] In rejecting claim 15 – the features of which are now incorporated in claim 1 – you cite paragraph 30, lines 4-8 of Smith, which simply mention the well known HTML language. As you note, HTML can define how elements are displayed. I submit though, that the mere mention of the well-known HTML language does not teach or suggest a group of services which both formats content and arranges graphical objects. A language is not a "group of services."

Proposed Amendments

[0008] Please see the attached Appendix of Proposed Claim Amendments. I would like to discuss your opinion regarding the proposed amendments in light of the currently cited references.

INFORMAL COMMUNICATION: Please do not put in the file

[0009] Thank you in advance for scheduling time for this interview. I look forward to discussing this with you.

Respectfully Submitted,

Dated: September 25, 2008

By: _____

Robert C. Peck
Reg. No. 56826
(206) 315-4001 x219
robp@leehayes.com
www.leehayes.com

My Assistant: Megan Arnold
(509) 324-9256 x270
megan@leehayes.com

INFORMAL COMMUNICATION: Please do not put in the file

Appendix of Claims with Proposed Amendments

1. (Currently Amended) A programming interface embodied on one or more computer readable storage media having computer-executable instructions for performing steps comprising:

generating graphical objects using a first group of services;

formatting content using a second group of services, wherein the second group of services arrange the graphical objects;

creating components of the graphical objects using a third group of services;

binding elements to data sources, data source classes, and data specific implementations of data collections using a fourth group of services, wherein the fourth group of services further handle exceptions in data entry;

using a common markup language to map classes and properties specified in the markup language to an instantiated tree of objects across the first group of services, the second group of services the third group of services, and the fourth group of services; and

integrating the first group of services, the second group of services, the third group of services, and the fourth group of services using a

INFORMAL COMMUNICATION: Please do not put in the file
consistent programming model and consistent services across the three
service groups.

2. (Canceled)

3. (Canceled)

4. (Previously Presented) A programming interface as
recited in claim 1, wherein the first group of services, the second group of
services, the third group of services, and the fourth group of services
share a common event system.

5. (Previously Presented) A programming interface as
recited in claim 1, wherein the first group of services, the second group of
services, the third group of services, and the fourth group of services
share a common property definition system.

6. (Previously Presented) A programming interface as
recited in claim 1, wherein the first group of services, the second group of
services, the third group of services, and the fourth group of services
share a common input paradigm.

INFORMAL COMMUNICATION: Please do not put in the file

7. (Previously Presented) A programming interface as recited in claim 1, wherein the first group of services, the second group of services, the third group of services, and the fourth group of services share a common system for nesting elements associated with a particular group of services within elements associated with another group of services.

8. (Original) A programming interface as recited in claim 1, wherein the first group of services includes a service that determines an appearance of the graphical objects.

9. (Original) A programming interface as recited in claim 1, wherein the first group of services includes a service that determines a behavior of the graphical objects.

10. (Original) A programming interface as recited in claim 1, wherein the first group of services includes a service that determines an arrangement of the graphical objects.

INFORMAL COMMUNICATION: Please do not put in the file

11. (Original) A programming interface as recited in claim 1, wherein the first group of services includes a plurality of nested elements that define the graphical objects.

12. (Original) A programming interface as recited in claim 1, wherein the graphical objects are comprised of one or more elements defined by vector graphics.

13. (Original) A programming interface as recited in claim 1, wherein the first group of services can define window properties in a markup language without launching a new window.

14. (Original) A programming interface as recited in claim 1, wherein the first group of services generate a user interface containing a plurality of graphical objects.

15. (Canceled) A programming interface as recited in claim 1, wherein the second group of services arrange the graphical objects.

16. (Original) A software architecture comprising the programming interface as recited in claim 1.

INFORMAL COMMUNICATION: Please do not put in the file

17. (Currently Amended) An application program interface embodied on one or more computer readable storage media having computer-executable instructions for performing steps comprising:

generating graphical objects using a first group of services;

formatting content using a second group of services, wherein the second group of services arrange the graphical objects;

creating components of the graphical objects using a third group of services;

binding elements to data sources, data source classes, and data specific implementations of data collections using a fourth group of services, wherein the fourth group of services further handle exceptions in data entry;

wherein the first group of services, the second group of services, the third group of services, and the fourth group of services are integrated via:

sharing a common programming model; and

using a common markup language across the three services to map classes and properties specified in the markup language to an instantiated tree of objects.

18. (Canceled)

INFORMAL COMMUNICATION: Please do not put in the file

19. (Original) An application program interface as recited in claim 17, wherein the third group of services includes services to generate geometric shapes.

20. (Original) An application program interface as recited in claim 17, wherein the second group of services includes arranging a plurality of data elements.

21. (Original) An application program interface as recited in claim 17, wherein the first group of services includes:

a service that determines an appearance of a graphical object; and

a service that determines a behavior of the graphical object.

22. (Original) An application program interface as recited in claim 17, wherein the first group of services includes a service that defines window properties in a markup language without launching a new window.

23. (Currently Amended) A computer system including one or more microprocessors and one or more software programs, the one or more software programs utilizing a programming interface to request

INFORMAL COMMUNICATION: Please do not put in the file

services from an operating system, the programming interface including separate commands to request services consisting of the following groups of services:

a first group of services for generating graphical objects;

a second group of services for formatting content, wherein the second group of services arranges the graphical objects;

a ~~second~~ third group of services for creating components of the graphical objects; and

a ~~third~~ fourth group of services that bind elements to data sources, data source classes, and data specific implementations of data collections, wherein the ~~third~~ fourth group of services further handle exceptions in data entry;

the first group of services, the second group of services, and the third group of services are integrated by sharing a common programming model, consistent services and using a common markup language to map classes and properties specified in the markup language to an instantiated tree of objects across the first, second, and third group of services.

24. (Currently Amended) A computer system as recited in claim 23, wherein the first group of services includes:

a service for defining an appearance of the graphical objects; and

INFORMAL COMMUNICATION: Please do not put in the file
a service for defining an arrangement of the graphical objects.

25. (Currently Amended) A computer system as recited in claim 23, wherein the second third group of services includes services to generate a plurality of geometric shapes.

26 - 34. (Canceled)